Consultative Committee for Space Data Systems

REPORT CONCERNING SPACE DATA SYSTEM STANDARDS

SPACE DATA SYSTEMS OPERATIONS WITH STANDARD FORMATTED DATA UNITS:

SYSTEM AND IMPLEMENTATION ASPECTS

610.0-G-5

GREEN BOOK

FEBRUARY 1987

NOTE

This Green Book contains many valid views of SFDU operations, and should prove to be useful to agencies and SFDU implementors as an overview of the main concepts. It should be viewed as a planning document of a concept that is currently in the process of evolution and development. An agency may adopt its own unique conceptual views of SFDUs provided that agency SFDU implementation conform to the SFDU data definition and data formatting interchange standards specified in the applicable CCSDS Panel 2 approved recommendations (Blue Books).

AUTHORITY

```
*******************************************************
*     Issue:          Green Book, Issue-5             *
*     Date:           February 1987                   *
*     Location:       CCSDS Plenary Meeting           *
*                     November 17-21, 1987            *
*                     Frascati, Italy                 *
*******************************************************
```

This Report reflects the consensus technical understanding of the Panel 2 members representing the following member Agencies of the Consultative Committee for Space Data Systems (CCSDS):

- o Centre National D'Etudes Spatiales (CNES)/France
- o Deutsche Forschungs-u. Versuchsanstalt fuer Luft und Raumfahrt e.V (DFVLR)/ West Germany
- o European Space Agency (ESA)/Europe
- o Indian Space Research Organization (ISRO)/India
- o Instituto de Pesquisas Espaciais (INPE)/Brazil
- o National Aeronautics and Space Administration (NASA)/USA
- o National Space Development Agency of Japan (NASDA)/Japan

The following observer Agencies also technically concur with this document:

- o Department of Communications, Communications Research Center (DOC-CRC)/Canada
- o Science and Engineering Research Council/Rutherford Appleton Laboratory (SERC/RAL)/UK

This document is published and maintained by:

CCSDS Secretariat
Communications and Data Systems Division
Code-TS
National Aeronautics and Space Administration
Washington, DC 20546, USA

FOREWORD

This document is a report on space data systems operations with Standard Formatted Data Units (SFDUs). It has been prepared by CCSDS Panel 2, Standard Data Interchange Structures (SDIS). The reader should refer to the glossary to ensure a proper understanding of the terminology used herein.

The document explains the rationale for operations with SFDUs, the initial operational requirements for SFDUs, and the major benefits to be expected from operations with SFDUs.

The concept of "open" systems versus "closed" is emphasized and several space mission scenarios, each "opened up" at different points, are presented.

To define operations with SFDUs means to describe how SFDU data format definitions are created, how instances of SFDUs are created, stored, transferred to their users and finally interpreted by them to obtain the data they are interested in, which is a goal of a space data system.

Each of these phases of operations with SFDUs requires implementation considerations. These considerations are essential in order to proceed to concrete applications of the concept. The implementations of the concept will evolve as practical experience is gained and technology advances.

The reader of this document is encouraged to become familiar with the terminology used in the ISO Reference Model of Open Systems Interconnection (OSI).

Documentation related to SFDUs and SFDU concepts prepared by the CCSDS comes in four colours:

Blue Books: Recommendations which reflect member Agency consensus on future technical direction and provide a basis for formal Agency-internal standards;

Red Books: Draft Recommendations which have been approved by the SDIS Panel and are ready for extensive, formal, Agency-wide review.

White Books: Internal documents which serve as a baseline for further panel-internal discussion and contain the first drafts of items that may become future Recommendations. Starting in 1987 White Books are not referenced in the official CCSDS documents (Blue, Red, and Green) which are provided with a CCSDS Document No.

Green Books: Reports formulated wholly within a panel at CCSDS that typically provide the rationale for recommendations and background information on related problems and issues;

In addition, the CCSDS has a procedures manual (Reference [l]) that describes its composition and goals, defines various internal administrative protocols and procedures, and describes protocols and procedures that facilitate the incorporation of its Recommendations into Agency data systems.

The following documents on SFDUs and SFDU concepts have been produced.

1. Space Data Systems Operations with Standard Formatted Data Units: System and Implementation Aspects (Green Book, Issue 5, February 1987; this document) describes the purpose of SFDUs, presents scenarios of the use of SFDUs to exchange data, and discusses design considerations.

2. Standard Formatted Data Units - Structure and Construction Rules (Red Book, Issue 2 February 1987) - describes the abstract structure, type-length-value encoding, and concrete syntax for SFDUs and supplementary data.

The following work leading to documents is underway:

1. Requirements for Data Descriptive Languages - specifies requirements for data format definition languages derived from analysis of existing data formats and from abstract data models, and includes assessments of currently available DDLs.

2. Requirements for SFDU Format Catalogs and Data Dictionaries - describes the automated facilities necessary to support the use and reuse of common data format specifications.

3. SFDU Control Authority Procedures - identifies current Control Authorities, describes their responsibilities, and establishes guidelines to facilitate the exchange of information.

4. SFDU Exchange Protocols - Specifies the protocols to be used for SFDU exchanges in heterogeneous environments. These are, for example, communication protocols, protocols to go from logical levels to physical levels, and standard representation of data types (e.g., integer, floating point).

DOCUMENT CONTROL

Document No.	Title	Date	Status/Remarks
CCSDS 610.0-G-5	Report Concerning Space Data System Standards: Space Data Systems Operations with Standard Formatted Data Units: System and Implementation Aspects. Green Book, Issue-5	February 1987	Current Issue

CONTENTS

CONTENTS (Continued)

REFERENCES

[l] "Procedures Manual for the Consultative Committee for Space Data Systems," CCSDS Document, Issue-l, August 1985, or later issue.

[2] "Recommendation for Space Data Systems Standards, Standard Formatted Data Units-- Structure and Construction Rules," Document No. CCSDS 620.0-R-2, Red Book, Issue-2, February 1987.

1. RATIONALE

1.1 STATEMENT OF THE PROBLEM

In space data systems applications, a principal objective is to make many types of data readily available to users. Frequently it is difficult for users to know what data are available, or to locate adequate descriptions of the data. Even after the data and adequate definitions are located, it is often costly for users to write interface programs to interpret the data and prepare them for further analysis by applications software.

The above situation comes about from two characteristics of most space data:

(a) The received data do not contain any information about their format nor their representation, and therefore their linkage to an external format description can be easily lost.

(b) The data formats and representations are not defined in any standard computer-interpretable way, so that even if the data format is known, manual translation is required to interpret it.

In typical space data systems projects, systems engineers write Interface Control Documents defining all the data which the project requires to cross each of its interfaces. This works satisfactorily if the interfaces are sufficiently simple and few in number. However this traditional approach has often forced agencies which provide services for several projects to deal with a proliferation of data formats and structures.

When projects have to "open up" their data systems due to expansion of their user population, or when a service provider offers service to many different projects, or when users need data from several projects (especially when the projects no longer exist and the data have not been submitted to an archive, as is the case for much still-valuable space data), the data frequently became difficult or impossible to interpret, because of characteristics (a) and (b) above.

Currently used technical approaches to providing open data systems have not addressed this problem. SFDUs are a data structuring technology and operational approach which promises to help solve this problem.

The fundamental purpose of the SFDU is to facilitate the exchange of scientific and technical data. The SFDU accomplishes this purpose by:

1. Facilitating the manual, and enabling the automated, exchange of space data in a heterogeneous data processing environment.

2. Reducing the time, effort, and costs associated with having to accommodate many different formats serving the same or similar functions. These redundancies are encountered as a consequence of (l) the uncontrolled proliferation of redundant formats, (2) the software development associated with the support of redundant formats, and (3) the required conversions to new computers as a function time.

3. Extending the useful lifetime of the data by assuring that the documentation of its form and content is recoverable for as long as the data are of potential value.

4. Allowing the automatic "parsing" of the interchanged data (in order to reduce errors, increase speed, reduce effort, and reduce costs.)

5. Facilitating, and in some cases enabling, the management of the anticipated large volumes of data, and their associated data descriptions, from future space projects.

6. Permitting the efficient establishment and preservation of audit trail information.

To accomplish these purposes, SFDUs and/or the SFDU concept must be able to:

1. Carry any type of digital data.

2. Accommodate data that is already in a standardized format, but which is different from, and was not originally compatible with the present SFDU guidelines.

3. Provide a method of allowing for the future evolution of the SFDU standards, without invalidating past versions.

4. Minimize data processing for processing-time-limited applications.

5. Provide for the global recognition of an SFDU instance.

6. Provide for the global identification of the format of an SFDU instance.

7. Provide for the physical separation of an SFDU instance from its format definition without introducing ambiguity.

8. Provide the ability to locate and delimit any structural element of an SFDU

9. Provide for the identification of common classes of data to support processing efficiency.

10. Allow for the inclusion of appended data when desired.

Additional attributes felt to be useful, but not necessary for all SFDUs, are to:

1. Provide the ability to easily identify the beginning and the end of an SFDU.

2. Provide the ability of an SFDU instance to carry its own format definition.

3. Provide for an Error Correcting Code file to be used if desired.

4. Identify the sender of the SFDU.

5. Identify the recipient of the SFDU.

6. Provide for the identification of the specific SFDU instance being sent.

7. Provide fields for the date, and time, that the sender placed the request to send the data.

8. Provide a field for the handling of priority of the SFDU.

9. Provide fields for describing the intent of the SFDU (e.g., info, action, etc.).

10. Provide fields for action requests from the recipients (e.g., acknowledgement requested, etc.).

1.2 SFDU CONCEPT

The SFDU concept defines interprocess data objects whose formats are described in a standard way for ease of identification and interpretation. Each SFDU is an individual conceptual object, consisting of a standardized label and data content, which is sent from an "originator" to a "recipient". This concept is depicted in Figure 1-1.

Each SFDU instance contains a globally interpretable unique data format identifier in the label portion which refers to a unique data format definition written in a formal language. The definitions are available from the Control Authority (CA) that registered the specific format. A data interpretation program can be developed manually from the formal data definition; or a general purpose interpreter can be used with the formal data definition to provide automated access to the data.

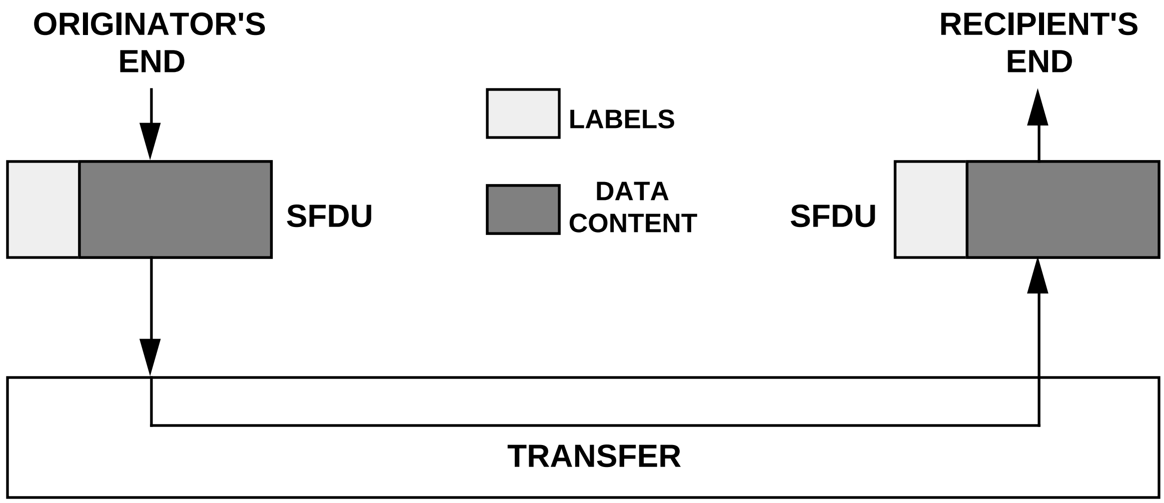

Figure 1-1. SFDU Interchange

The SFDU concept contains five main elements:

(a) SFDU data format definition:

The data format definition of an SFDU is specified in a data descriptive record (DDR) using a formal language capable of being interpreted automatically by a computer. The user does not need to have a programmer read a definition of the format in a human language and write a computer program to interpret the data. This allows an application program to consider an SFDU as a high-level structured data object, and be independent of variations in the bit-level detail, i.e., exhibit data independence. The data format definition of an SFDU may be obtained from the CA that registered the DDR. The responsible CA is identified within the label part of the SFDU. The CA maintains a data dictionary containing information describing its SFDU data format definitions. This makes it easy to locate data definitions and descriptions of interest.

(b) SFDU instance generation:

As needed, originators generate SFDU instances which conform to the appropriate SFDU data format definitions. An originator may be associated with a primary data source such as a space instrument, or with a secondary source such as an archive. SFDUs may be generated systematically, such as when a space center is transmitting data to another center or end user, or it may be a catalogue of interest to the user. The generation of a label which, as a minimum, identifies itself as an SFDU and gives the label version, the CA, the data object class, the DDR ID, and the length of the SFDU instance.

(c) SFDU Service:

Each source of SFDUs provides access to its SFDU services.

For a concept where a CA agent holds SFDU definitions/descriptions, such a service offers the following operational features:

- A capability to accept requests for SFDU data format definitions by means of a request procedure (possibly standardized).

- A capability to deliver these formats in the form of a specific SFDU instance as explained above. See (a).

- A capability to annotate deliveries so that they can be associated with particular requests.

For a service agent that also has the ability to deliver SFDU instances:

- A capability to accept requests for SFDU instances in the defined formats which have been previously requested and delivered.

- A capability to deliver SFDU instances in response to the previous requests.

SFDU instances are handed over to a communications system which guarantees to deliver them semantically unaltered to the SFDU requester.

Requesters of SFDU data format definitions and SFDU instances may use a communication system which is more suitable for relatively low data rates and interactive procedures, provided the volume of instances requested is not excessive.

(d) SFDU Communications and storage:

Communications and storage domains are theoretically transparent to SFDUs.

The data content created at the SFDU originating system by an application process is delivered in a semantically unaltered form to the peer level of the recipient system regardless of the transmission and storage means. For communication of SFDUs, in terms of the ISO Reference Model of OSI, SFDUs are originated by an application process and passed to the Application Layer (layer 7), thus establishing a communication request. Although what happens in the communication path is not directly of concern to the SFDU domain, communication protocols and physical storage representation are important aspects of any practical implementation of the SFDU concept. It is also to be noted that syntactic transformations or conversions may take place at the Presentation Layer leaving unaltered the semantic content of an SFDU. An archive may contain data in any internal form so long as the data presented to the recipient on retrieval from the archive is in its registered SFDU format.

More specifically, when a requester of SFDU instances receives them via exchange of a physical storage medium (e.g., magnetic tape, optical disk), the mapping of the SFDU instances to the physical structure of the medium, and the characteristics of the medium required for interchange, must also be provided. One approach to solving this problem is to physically attach a message containing the additional information to the exterior of the physical medium. This information is then used to manually configure the system input/output calls used to access the physical medium. Once the logical structure has been extracted from the physical structure, the automatic SFDU interpretation using the DDR can begin. This interpretation must, however, provide for the automated translation of low-level data structures, such as floating point words, to a representation useable in the recipient's system.

(e) SFDU interpretation:

When an SFDU instance is received, it can be interpreted automatically if it contains a unique identification of its format, which is defined in a formal language. The user does not need prior knowledge of the format of the data; the specific format is identified through the CA and DDR identifier (ADI), within the SFDU received. This makes the SFDU instance potentially accessible to all users.

It is to be noted that data format definition contained in the DDR and instances of this format may be stored in physically separated locations.

The SFDU also carries CCSDS/SFDU identification, through the control authority identifier, to facilitate operations in the current environment, which is not fully compatible with the OSI 7-layer communications model. It is expected that general purpose interpretation software will be developed that will use the DDR, written in an approved data descriptive language (DDL), to access the data and present it to user applications programs for further analysis.

1.3 USES AND BENEFITS OF SFDUs

Some space missions may have to be supported by different organizations. This involves exchange of data between these organizations. With respect to the final utilization of data or of a service, several examples of applications can be given:

- Scientific projects, where a user community receives data collected by some experiments which constitute the payload of the spacecraft concerned.

- Scientific projects, where data are exchanged among members of the user community in an unscheduled manner.

- Application projects where a user community receives, through a regularly scheduled service, some well identified data products. Typical of this category are, for example, the meteorological products delivered from meteorological satellites.

- Application projects where data systematically collected from a spacecraft are catalogued, are archived into a data base, can be retrieved, and are made available to customers on request through an organized service. Typical of this category are earth observation data.

The technique of using SFDUs is intended to improve the way in which data are exchanged and is particularly indicated when one or more of the following conditions apply:

- Cross-support is needed between different organizations for mission operations.

- The user community is made up of several members each of whom is entitled to get data products and each of whom intends to further analyze and process the data on their own computer facilities.

- The user community is open, in the sense that a new member not known a priori may join the community at any time, even several years after data acquisition.

- Originators and recipients of the data do not belong to the same organization and are not physically resident on the same site.

- An organization is using common facilities to support several projects at the same time. Each individual project may or may not belong to the same agency organization.

In cases such as the above, the following benefits of using SFDUs may be expected as implementations of the concept evolve:

- A key feature of SFDUs is the capability for recognizing and unpacking the data content automatically, with computer software, by means of the format definition written in an agreed-to formal DDL. This would allow a user who needs to receive data from various locations or from various disciplines to invest only once in universal data interpreters or decoders. This software could be utilized by several different centers.

- SFDU data format definitions can be stored and retrieved automatically using data dictionaries. This would result in reduced proliferation of data formats, and improved data archiving and distribution services.

- This SFDU concept helps solve the management problem of interpreting the content of stored data, e.g., in a tape library, and retrieving them a long time after they were archived. Frequently, it is difficult to find an appropriate description of the format, and the people responsible for the implementation of the archive have left. Under the SFDU concept the formatting aspects of the problems would be brought under control.

- Scientists often need to make correlations of data from multiple sources and from different disciplines. The use of SFDUs would mark a substantial step forward toward a satisfactory solution of this problem of great interest for many scientific communities.

- Widespread implementation of SFDUs by space agencies will encourage multi-national space missions by facilitating agency cross-support with potentially significant cost savings and allowing improved scientific cooperation.

- Agencies which provide service to several projects can use the same standardized hardware/software.

- The SFDU concept may also be of value to other communities outside the space data community.

2. SPACE MISSIONS WITH SFDUs

2.1 INTRODUCTION

This section gives a broad operational description of how SFDUs would be used in typical space missions. The SFDU focus within CCSDS is on mission concepts which involve multi-national participation.

The following scenarios of space missions give some idea of possible utilizations of SFDUs. They are described in order to provide examples of the general concepts of SFDUs outlined in Section 1.

In order to illustrate potential SFDU applications for space missions, an extremely simplified mission model has been developed. Its only purpose is to demonstrate various degrees of "openness" of a space project achievable using SFDU techniques. Connections between functions which are unlikely to become access points to the system have been intentionally omitted. The scenarios presented may be combined into a completely "open" mission model.

2.2 "CLOSED" VERSUS "OPEN" SYSTEMS

Definitions

A "closed" system is a system with its own private data formats and protocols which it uses internally and does not share on a broader basis.

An "open" system is one which uses publicly available formats and protocols so that, in theory at least, anyone can communicate with the "open" system by following the "open" system standards. It should be noted that "open" does not imply an uncontrolled or unrestricted access to the data.

SFDUs in "Closed"/"Open" Systems

In space data systems the portions of a system which are "open" or "closed" for a particular project depend on project requirements and implementations.

For example, the "closed" part may comprise the project data system for generation at the source (i.e., onboard the spacecraft), mission command and control, and data capture in a holding area. In this "closed" part private formats and protocols could be used; these could, however, also be SFDUs and the protocols used could be standardized ones not accessible to external users.

Agencies' multi-mission systems (facilities which support several projects at the same time) are in this context "open."

The "open" part would consist of the user access interface to the data capture system and the data processing and analysis system, and data archives. The "open" part would by definition use SFDUs and publicly available protocols.

2.3 SFDU SCENARIOS

2.3.1 SCENARIO A: ACCESS TO MISSION ARCHIVES ("A"s in Figure 2-1)

End users (e.g., scientists, spacecraft designers) may request data from a past mission. If such data exist, both the data and their format description in a DDL can be obtained from the CA responsible for their archive and will be delivered as SFDU instances. (Note: it is assumed that the archive is also a CA.)

If the end user during or after the mission generates data which are considered worth retaining, they may be handed over to an archive (e.g., a space science data center) in the form of SFDUs. The SFDU instances will thereafter be maintained by the archive; the related SFDU data format definitions will be maintained by the responsible CA.

2.3.2 SCENARIO B: DATA PRODUCTS ACCESS ("B"s in Figure 2-1)

The space project allows access to some portion of its data processing facilities (e.g., an observatory) to end users and archives which are not necessarily part of the project.

End users and archives which receive data from space data processing centers would be able to interface with these facilities with the same tools, provided all of them are able to deliver SFDUs as output data structures.

A growing trend in missions is to include multi-disciplinary scientists/users. In addition there are many cases where a new project wishes to utilize the existing facilities at a user's institution for science processing.

2.3.3 SCENARIO C: ACCESS TO SPACECRAFT TELEMETRY ("C" in Figure 2-1)

In a space mission which is supported by several agencies it is conceivable that telemetry data acquired by the facilities of one agency be processed using facilities of another agency or of an end user (e.g., Principal Investigator). In this case it might be useful to provide the telemetry in the form of SFDUs after conditioning (e.g., frame-synchronization, quality test, or even calibration).

Data conditioning facilities may be required to send data to many different data processing centers and, on the other hand, a data processing center may receive input from a variety of data acquisition and conditioning facilities.

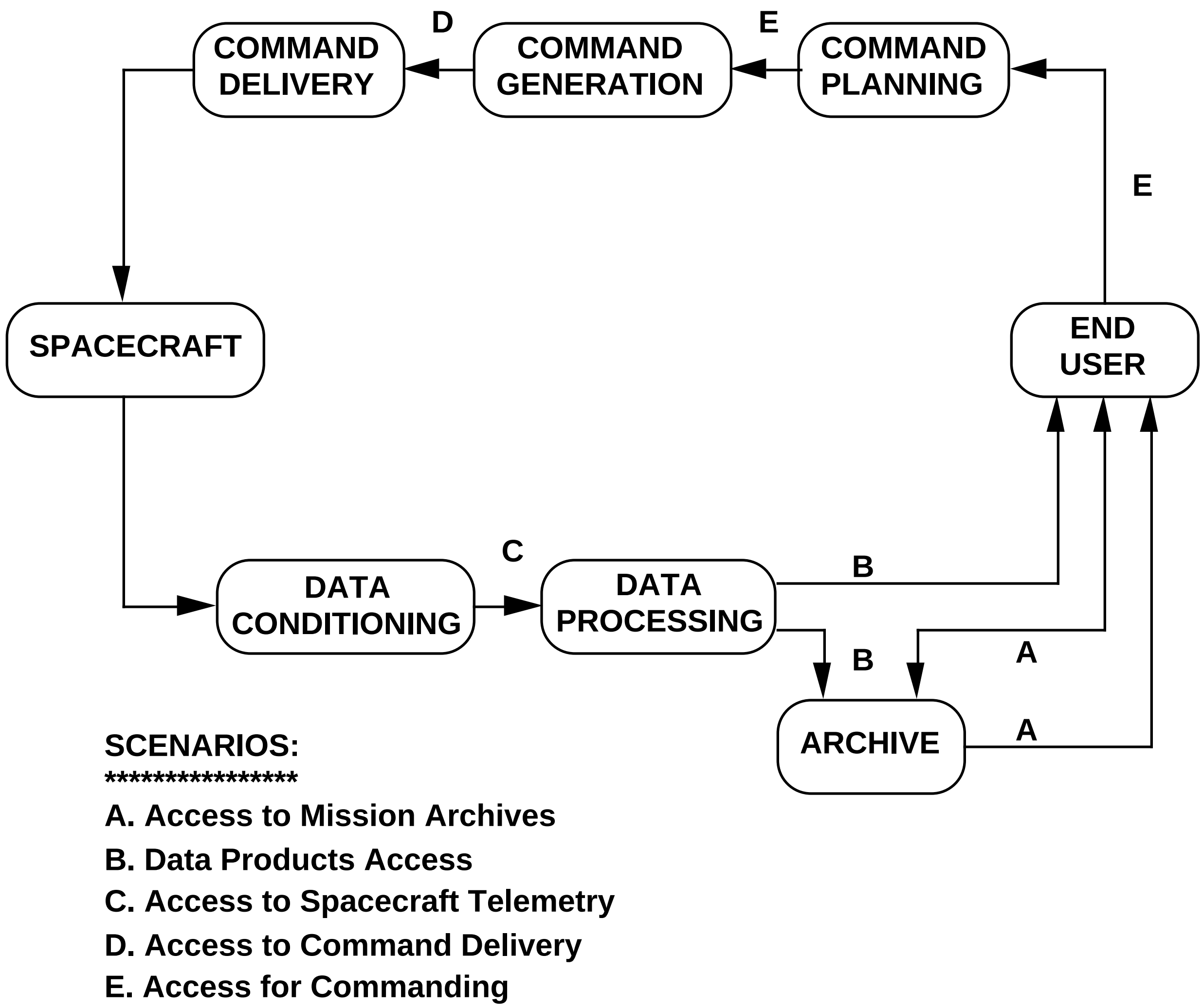

Figure 2-1. Information Flow in a Typical Space Data System

SFDU structures would provide a uniform logical interface and thus considerably reduce investments required to implement interfaces to different data acquisition systems and their particular data delivery formats.

2.3.4 SCENARIO D: ACCESS TO COMMAND DELIVERY ("D" in Figure 2-1)

The situation for cross-support for commanding is similar to the situation discussed in Scenario C.

A command generation facility of an agency (e.g., a control center) may require several ground stations of different agencies to radiate the commands to a spacecraft.

If the command generation facility presented commands as SFDU instances, an interpreter at the delivery facility could (using the format definition for such commands) extract, e.g., spacecraft identification, radiation time, orbit predictions and feed this information into the relevant subsystems. The command generation facility need not care about peculiarities of individual ground stations but only has to provide sufficient information in a well defined SFDU.

Similarly, individual ground stations that accept SFDUs need not be concerned about peculiarities of different command generation facilities.

In general the SFDU data format definition giving the command format would be known and verified beforehand and not be included with the commands for dynamic handling.

2.3.5 SCENARIO E: ACCESS FOR COMMANDING ("E"s in Figure 2-1)

End users (e.g., Principal Investigators) may be allowed to command their instrument onboard a spacecraft under their own responsibility. In this case, the access to the command system could be "allowed" as shown in Figure 2-1.

Various instrument-specific command planning facilities present their requests for commands in the form of an SFDU to the command generation facility, which integrates these commands with others and forwards them for transmission to the spacecraft using Scenario D.

It is also possible that some end users share a command planning facility and transmit to it their commanding requirements in a standardized form (i.e., SFDUs) for coordination, optimization and translation into more spacecraft oriented terms.

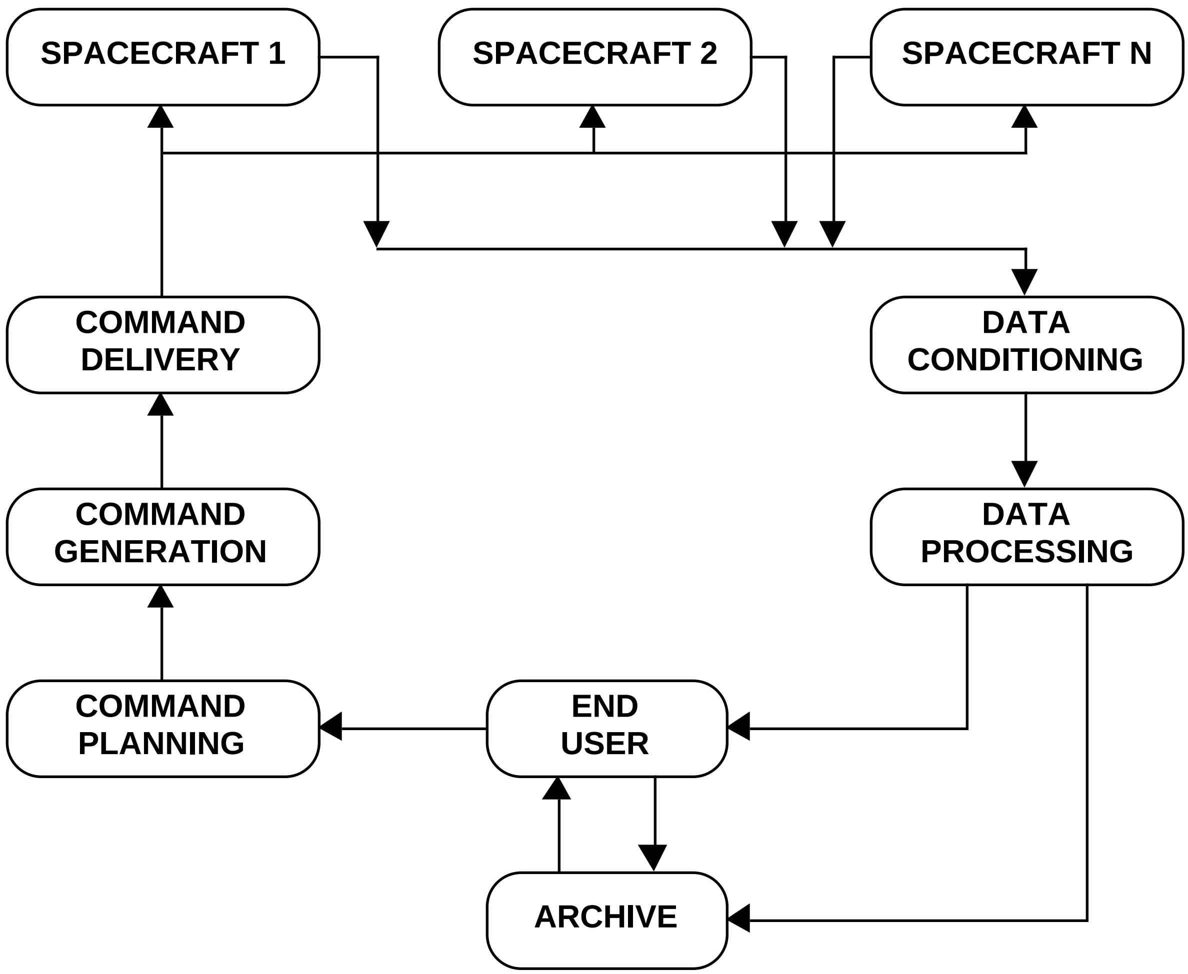

Figure 2-2. Multi-Mission Service (Scenario F)

2.3.6 SCENARIO F: MULTI-MISSION SERVICE (Figure 2-2)

Agencies which provide service to several projects at the same time will find the SFDU a useful concept for all message transactions (telemetry, command, radiometric, system status, schedule, etc.) with the supported projects.

The ground systems which support multi-disciplinary missions such as the Spacelab and the Space Station, where individual instruments are changed out and the vehicle reflown, are other examples where the SFDU concept can be of benefit.

3 CONCEPT OF OPERATIONS: SFDU SYSTEM CONSIDERATIONS

3.1 INTRODUCTION

In this Section we discuss some of the details of how SFDUs are defined, how instances are created, how they are stored and transferred among users, and how they are interpreted. This discussion will avoid abstract concepts and, instead, present SFDUs in terms of concrete examples. (The examples are not intended, however, to restrict the use of other techniques for defining and exchanging SFDUs.)

An overview of the life cycle of an SFDU is shown in Figure 3-1. This is a case study of one particular type of SFDU; other scenarios are possible. Life begins with the definition of the logical and physical structure of the data to be contained in the SFDU. The data format definition is expressed in a formal language so that it can be registered with a CA and distributed to SFDU users. The next step is to create instances of data units which conform to the specified structure and include the necessary identification; instances may be generated directly from source data or from archived data. SFDU instances may also be stored for short-term project use or archived for longer-term access by other users. SFDU instances may also be created for distributing data electronically over communications networks or via storage media (e.g., tapes or optical disks). The final step is to interpret the SFDU, which means to verify the identification of the SFDU and extract the data it contains in a useful form.

3.2 SFDU STRUCTURE AND CONTENT

The invariance of the conceptual structure of an SFDU is an important key to its utility. Physical representations in computer memory, on storage media, and in electronic communications may differ. The conceptual structure, however, must be maintained. For this discussion, the physical structure of an SFDU is assumed to be a sequence of octets upon which a conceptual structure is superimposed.

The top layer of the conceptual structure divides all SFDUs into conceptual parts consisting of a label and the data content. These two parts contain the following elements:

```
SFDU:
   Label:
       Control Authority Identifier
       Version Identifier
       Class Identifier
       Data Descriptive Record Identifier
       Length of Data Content Field
   Data Content:
       User Data Structures
```

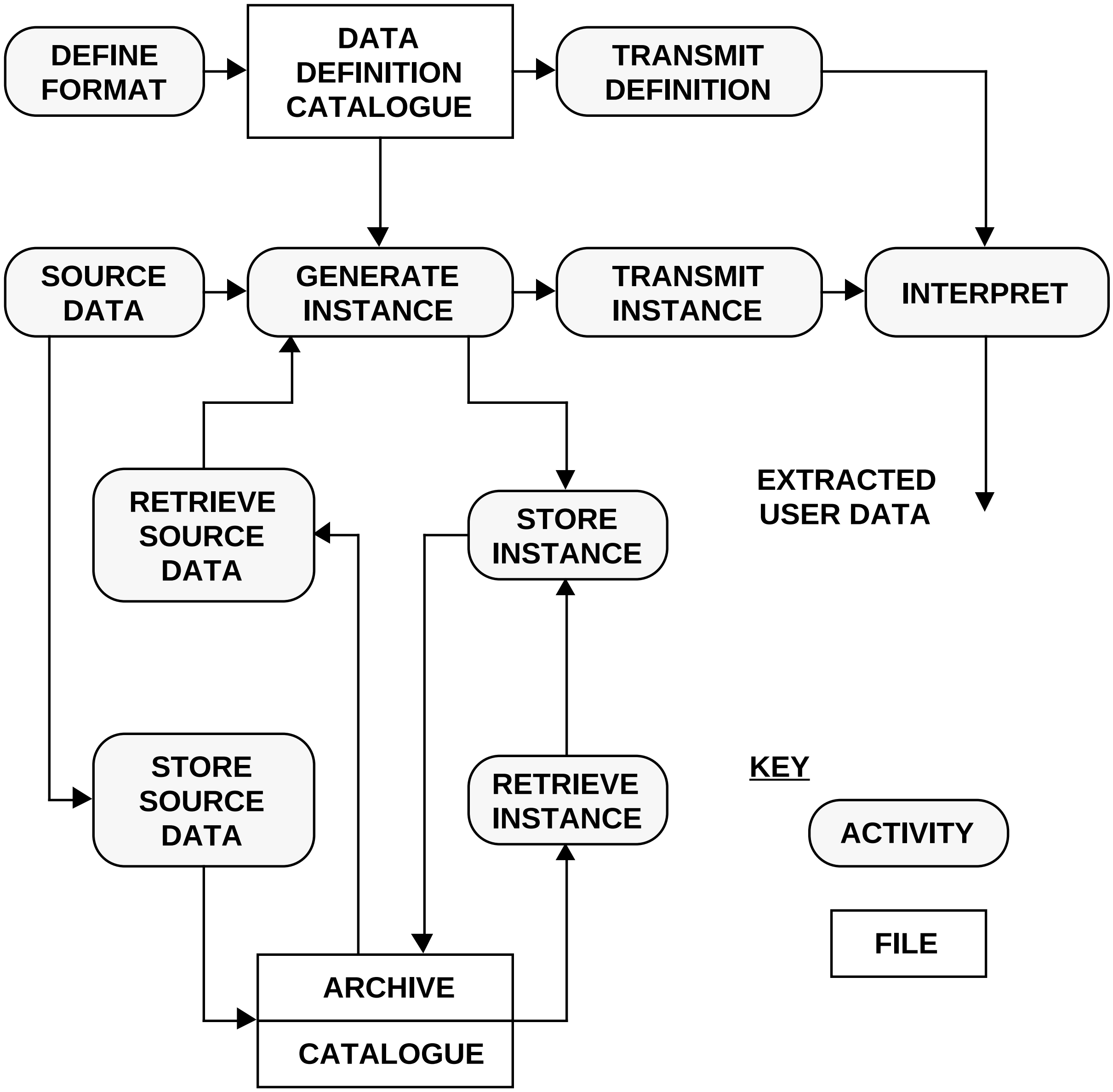

Figure 3-1. Diagram of SFDU Life Cycle

This structure is compatible with Type-Length-Value (TLV) encoding, where the type field contains the CA identifier, version identifier, and data format and description (DDR) identifier; the length field contains the length of the data content field; and the value field contains user data structures. This encoding approach is the recommendation for the basic structure of an SFDU, and is described in detail in Reference [2]. In this recommendation, the type field includes a class identifier and the value field (Data Content) of an SFDU is made up of additional TLV structures. An example of this conceptual structure incorporating a secondary label (or header) together with two sets of user data is shown in Figure 3-2.

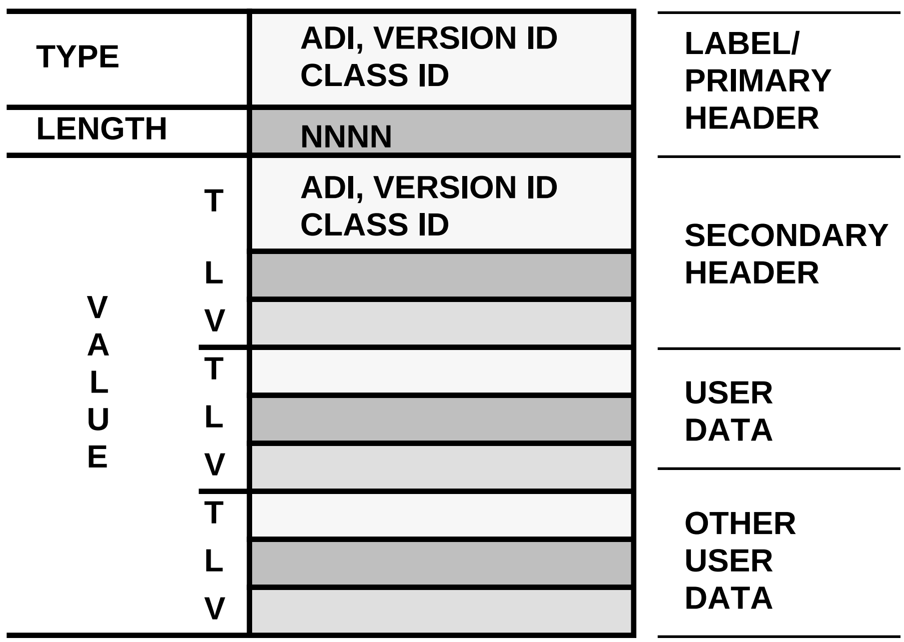

Figure 3-2. Sample TLV Encoding of an SFDU

Note that the value field of this SFDU, which has a length of NNNN, is made up of 3 other TLV Objects. The first serves the role of a secondary header, while the remaining two contain user data. In each TLV Object, the version number (ID) uniquely identifies the abstract syntax of the label (type-length fields), the class ID identifies the type of value field and the ADI (Authority and Data Descriptive Record ID) uniquely identifies the CA and DDR ID needed to obtain the format definition of the value field from the CA. This abstract syntax of the conceptual structure of an SFDU, based on TLV encoding, is standardized to allow universal interpretation.

Unique instance identifiers, time stamps, and other tags required to distinguish SFDU instances of the same DDR are included in supplementary headers as required by the application. In addition, ancillary data such as calibration factors, references to other data, and investigator's notes may be included in supplementary headers. However, guidelines for defining widely useful secondary headers may be established as well as standard definitions for common items such as time stamps.

The structure of the user data is left unspecified by CCSDS although, in the future, some additional useful data structures may be identified. The only requirement is that the definition of the structure be completely specified in a formal language recognized by CCSDS to facilitate the exchange of data within the user community. A discussion of data format definition techniques is presented in the next section.

3.3 SFDU DATA FORMAT DEFINITION

The key to "open" exchange of SFDUs is that their structure be defined in terms that have an agreed-upon meaning and interpretation. To ensure this, a formal specification is required for each unique SFDU format. This specification must be expressed in a DDL which is machine readable, and can be interpreted and processed automatically by supporting software.

Modern DDLs provide facilities for describing data in application oriented data types and data structure terms (e.g., real numbers, arrays, records, files, relations). Such application oriented data structures simplify the programming task and lead to higher software portability, maintainability and reliability. Currently, few space data systems make full use of these modern software data definition techniques.

As an example of data format definition using a programming language, consider the following Ada type definition of the SFDU label:

```
Type      CCSDS SFDU Standard_Label is
          | Record
          |    CTRL_AUTH : SFDU_CONTROL_AUTHORITY_IDENTIFIER;
          |    VERSION: SFDU_VERSION_CODE;
          |    CLASS_ID: SFDU_CLASS_IDENTIFIER;
          |    SPARES: SFDU_SPARE_OCTETS;
          |    DDR ID: SFDU_DATA_DESCRIPTIVE_IDENTIFIER;
          |    LENGTH: SFDU_VALUE_FIELD_LENGTH;
          | End Record;
```

This definition, which is given in a very high-level "abstract syntax," is not complete as it stands. Complete definitions for each of the components must be provided, down to their physical, bit-level representations and locations ("concrete syntax"). Each different concrete syntax of the label is a different version.

Note that the concrete syntax of a data structure used entirely within the closed system environment of a single machine normally is not specified by the user, but instead is left for the compiler to decide on. In an open systems data transfer environment however, the bit-level layout must be specified by the DDL (or resolved by use of a presentation protocol). Therefore, DDLs to be used in a SFDU context must allow bit-level syntax to be specified.

One DDL for defining all SFDU types would be ideal, but may not be possible in practice, since different DDLs may be appropriate for different objects or applications. The number of DDLs adopted should be controlled by the CCSDS and kept to a minimum.

SFDUs are not necessarily small and simple objects; in fact, they could have arbitrarily large size and complexity. The structure of SFDUs using TLV encoding enables the construction of arbitrarily large and complex data structures.

SFDU data format definitions should be stored using a data dictionary associated with the CA, initially under the local DDR ID as access key. Later on, more sophisticated schemes could be implemented.

3.4 SFDU INSTANCE GENERATION

SFDU instance generation is primarily a process of packaging user data into the predefined data format, including the required header information. Instance generation may be done at an early stage of the life cycle (see Figure 3-1) for storing collected data in the form of SFDUs, or it may be done at a later stage in response to database queries or to orders for processed data from an archive.

3.5 SFDU COMMUNICATIONS AND STORAGE

SFDUs can be exchanged between users either by electronic communications or by exchanging storage media such as tapes or optical disks.

SFDUs can be transferred as unstructured strings of octets. This mode of SFDU transfer could be supported either by transport protocols operating over communications networks, or by file management systems using physical storage media. With these approaches, a session protocol or media format would have to be determined and imposed by the user in order to read and write the SFDUs as structured objects.

A better method of transferring SFDUs electronically would be to transfer them as structured objects (i.e., label, supplementary labels, and structured user data) instead of as unstructured octet strings. File transfer protocols or electronic message handling protocols could accomplish this. These higher layer protocols free the user from concern with the concrete syntax of the physical data structure.

For data storage and retrieval, SFDUs may be organized and accessed either through a computer file management system or through a database management system (DBMS). File management systems treat SFDUs as sequences of octet strings. A more versatile and, sometimes, more useful approach is to store data under the supervision of modern database management systems (DBMSs). DBMSs provide many useful facilities which enable users to identify and locate "interesting" data. These may include data dictionaries, catalogs with cross-indexing capabilities, and extensive search facilities.

Note that, no matter how the SFDU transfer occurs - transfer protocol, file management system or DBMS - a physical transfer service must exist with the system to handle the protocol headers or file formats. The transfer service would extract the SFDU instance from the communication or storage format and pass it to the interpreter.

The above discussion of communications and storage refers just as much to SFDU data format definitions as to the actual SFDU instances, since the text of a data format definition can be packaged as an SFDU instance.

3.6 SFDU INTERPRETATION

SFDU interpretation refers to the process of recognizing the conceptual and logical structure of an SFDU instance, identifying its component parts, and extracting the information it carries. This process includes recognizing the label version and interpreting the concrete syntax of the CA and format identifier.

The format identifier allows an SFDU instance to be associated with the correct data format definition provided from the data dictionary of the designated CA. The data format definition in turn allows the correct interpretation of any additional labels and of the user data.

Application software can be constructed to interpret specific SFDU data formats. The data format definition of a particular SFDU, which is available from the responsible CA, is used to recreate the abstract data structure of the SFDU from the transmitted string of octets representing the SFDU instance. The handling of the low-level details of this reconstruction are carried out by the translation software that supports the formal DDL. This frees the user to concentrate on the logical relationships and meaning of the data.

In cases where the users who are exchanging SFDUs do not use the same DDL, a "translator" of data definitions from one language to the other may be required.

Powerful software which will handle storage and distribution of SFDUs corresponding to different DDRs can be constructed. Table-driven, DDL-specific processing techniques can handle a variety of SFDU data format definitions. Some general purpose SFDU interpreters can, therefore, be built. The required tables can be generated for a large number of SFDU data format definitions by automatically translating their specifications. This approach offers high reliability, and would quickly pay off the initial software investment.

3.7 OPERATIONS ON SFDUs

Defining abstract and concrete representations are only half the job of defining a new data object. To bring data objects to life their creators must also define and implement operations to manipulate, transform, and transfer them. In this section we describe the essential "primitive" operations necessary to create and transfer SFDU instances plus some convenient extended operations.

3.7.1 PRIMITIVE DATA OPERATIONS

Two fundamental kinds of operations on data objects are those that create new data objects from component parts, and those that extract component parts from objects. These operations may be supplied by a programming language (e.g., filling in an array) or they may be defined as separate functions and procedures. The advantage of defining separate functions and procedures is that, given an adequate set of such operations, applications software can be made completely independent of concrete data formats.

The essential operations on SFDU instances are to create or assemble them, and to interpret or unpack them into their constituent parts. Various degrees of automation can be associated with these operations. For example, the assemble operation may take a CA and format ID, appropriately formatted supplementary data (if required) and a set of appropriately formatted user data, and produce a complete SFDU instance. In contrast, the create operation may take unformatted results from a database query, pick an appropriate format for the data, and automatically fill in the CA, version, class, format identifier, and supplementary information. Similarly, the unpack operation may simply separate out the major components, whereas the interpret operation may recognize a number of common user data formats and directly extract more useful information.

3.7.2 EXTENDED "LIBRARY" OPERATIONS

Convenient, higher-level operations on SFDU instances that can simplify applications software considerably include operations to append data to an SFDU instance, to concatenate SFDU instances within a surrounding SFDU "envelope," and to extract data (SFDUs) from these structures. These and other common operations derived from practice enhance the utility of SFDUs by allowing users to easily compose data sets for exchange.

The append operation takes an existing SFDU instance and adds new information such as results from processing, instrument calibration data, correlative data from another source, or simply explanatory notes about the data. We assume that the original SFDU structure allows such extensions or that a new SFDU instance is created with the original as a component.

The concatenate operation takes a collection of existing SFDU instances and packages them together as a single data unit. This may be a convenient way to handle image data, for example. Each scan line may be represented as an individual SFDU instance. The composite picture can then be composed by concatenating the lines in a new SFDU instance.

The extract operation is included to ensure that the append and concatenate operations can be undone at a later time. This is essential to enable users to extract arbitrary subsets of data and freely combine data from different sources.

3.7.3 DATA EXCHANGE OPERATIONS

The essential operations for exchanging SFDU instances include the ability to read and write SFDU instances on physical storage media (tapes, disks, diskettes, video disks, etc.), and the ability to send and receive SFDU instances via communications media. Ideally, these operations would be integrated with the machine's standard input and output operations.

Ideally, the read and write operations would make all the details of the actual physical media transparent to the user. That is, they would completely hide the lower-level details of "internal" media formats, sector and record sizes, record and file marks, etc. Since the SFDU is an application-layer data object, it does not address these lower-level issues. Therefore, it is up to the physical media industry and the implementers of the read and write operations to provide a presentation-layer service to support the "open" exchange of SFDUs.

The send operation is conceptually very simple. It may require that a communications link be established first by a user, or it may be entirely automatic and make the necessary connect itself. The receive operation may or may not be directly initiated by a user. Larger computer systems provide for receipt of communications as an unattended user service. Personal computers, on the other hand, usually require users to set the machine to an "auto-answer" mode when waiting for calls.

3.7.4 COMMON HIGH-LEVEL EXCHANGE OPERATIONS

Many possible scenarios where SFDUs might be exchanged can be imagined. The more common of these will warrant standard exchange procedures that can be included as operations on SFDU instances. Operations to request a format definition, to register a new definition, and to browse a catalogue of format definitions are examples of transactions with a CA. Similar operations may be supported to browse through a data catalogue and request data from an archive.

We assume that common SFDU procedures for requesting and exchanging format definitions will be established. The request and register operations construct such data units and send them to the appropriate CA and wait for a reply. (The wait would presumably be a suspended background operation.) The browse operation is more of an interactive database querying activity which may or may not be an operation on SFDU instances. However, it involves exchanging information about SFDUs and could be built using SFDU instances to support the communications.

3.8 OPERATIONS WITH SFDUs

The SFDU concept, conceived as a mechanism to greatly facilitate the exchange and interpretation of data between heterogeneous systems in an open environment, is potentially much more significant since it may also revolutionize the way in which data are managed. Inherent in the concept is the notion of self documentation which potentially allows for highly automated archiving, cataloging, distribution, and presentation of data to user application systems.

Based on potential uses, several key types of SFDU transactions have been identified. The first broad class of usage is for data interchange, for which four major cases have been identified:

1. Ad hoc: Requests and responses (e.g., ad hoc DBMS queries and responses) are characterized by a need for one-time or infrequent use of a specific format. This is a good application for generalized formats which are totally self-describing (i.e., in-line DDR).

2. Frequently used formats: "Broadcast" or other commonly used formats (e.g., common ancillary data formats) offer an opportunity to optimize use of transmission media by storing the data formats at user sites and referencing required formats based on SFDU label format identification after transmission.

3. Media exchange: Physical transfer of media may be characterized by either of the above cases.

4. Real-time (or near-real-time) communications: Applications (e.g., payload data) which benefit from automated parsing of SFDUs (perhaps in a processing-time-limited environment).

The second broad class of usage is for archival storage, for which two major cases have been identified:

1. On-line: During the Space Station era it is anticipated that very large volumes of data will be catalogued and

archived. The ability to automate at least the initial categorization and archiving process can be based on SFDU information necessary for data interchange requirements.

2. Off-line: Much space data will be useful for extended periods of time if it can be interpreted long after it is recorded. This implies a need to preserve data formats, data definition language versions, etc., as well as, data for extended time periods.

The current lack of practical experience with SFDUs, coupled with the potential for revolutionary new ways of handling data, require that an evolutionary path for their development and implementation be recognized.

Annex C to this document, with three sections, titled "Initial Operations," "Intermediate Operations," and "Advanced Operations" provides a stimulus for thinking about the practical problems and benefits associated with the evolutionary implementation of SFDU concepts.

4 CONCEPT OF OPERATIONS: IMPLEMENTATION CONSIDERATIONS

In this section implementation considerations for SFDUs are discussed. These include questions of mapping the abstract data structures into concrete data structures and media, practical approaches to creating and maintaining data dictionaries and data archives, and aspects related to the development and use of data interpreters.

4.1 SFDU DATA STRUCTURE

An SFDU is a labeled data object. The label is the key to the interpretation of the data structure.

An idealized view of an "equivalence relation" for SFDU instances (i.e., when are two SFDU instances "the same?") would be that they conveyed the <u>same meaning</u> so far as an interpreter was concerned. For example, the floating-point values of telemetry parameters grouped in an SFDU would be seen as the same SFDU by an interpreter, although their physical representations in specific machines may be different.

The view that two SFDU instances are equivalent if and only if they have exactly the <u>same form</u> (i.e., identical bit-strings) seems to be too restrictive, because there are many useful encodings of data onto physical media which differ only in low-level details, such as header formats or number of logical records per physical record.

Two SFDU instances are equivalent if and only if: (a) they have the <u>same abstract syntax</u>, and (b) corresponding <u>data values are equal</u>. It would be possible in this view to have the "same" SFDU instance represented in two different versions of concrete syntaxes as long as they were equivalent in the above sense.

This would allow potentially different encodinqs of the same abstract data structure, e.g., to store SFDUs on a floppy disk or computer tape (which may require different physical data structures or blocking factors), or to send data electronically via electronic message or file transfer protocol (which may require different transparency protocols).

The label field denotes a specific SFDU data format definition. An SFDU instance of the indicated SFDU data format definition is a data object correctly interpretable according to the data format definition.

4.2 SFDU DATA FORMAT DEFINITION

A data format definition is considered to be an expression in a specified DDL.

An SFDU data format definition may be requested from its CA by giving its format ID. The data format definition could itself be transmitted as the data content of an SFDU.

4.3 SFDU INSTANCE GENERATION

When needed, an originator generates an instance of a defined SFDU, including the appropriate headers. The instance is then ready for local use, archive or transmission.

4.4 SFDU COMMUNICATIONS AND STORAGE

Although the communication and storage domains are transparent to SFDUs, the handling of SFDUs requires these aspects to be considered.

In the terminology of the ISO Reference Model of OSI (ISO 7498) and the CCITT Reference Model of Message Handling Systems (CCITT X.400), the SFDU instance generator is a user agent application process, and it transfers the SFDU instance to the recipient user agent using an agreed-to Application Protocol (layer 7). As part of the transfer, Presentation entities in the originator's and recipient's end systems provide necessary adjustments of the concrete syntax to be used. This "agreement" may of course be unilateral, in the sense that the originator's Presentation entity may simply select the concrete syntax to be used. This is particularly the style used for transfer via physical storage media such as magnetic tape. The selection is certainly "agreed-to" however, in the sense that the recipient's Presentation entity uses the same concrete syntax to interpret the bits received.

4.4.1 SFDU PHYSICAL REPRESENTATION

SFDU instances must be represented on physical media using specific mappings appropriate to the relevant physical media. Internal representations on local peripherals are largely irrelevant but for the purposes of data exchange the representation on computer compatible tapes and floppy disks is of interest.

In the long term, it is hoped that tape standards will be used, but, for expediency in the short term, this should be a matter of negotiation between the parties concerned with regard to the selection of a mutually acceptable convention. Examples of possible conventions in this area would be:

- ISO 1001-1979(E) Tape Standard
- ANSI-standard tapes
- IBM standard label tapes with a specific record structure or generated using an IBM utility
- DEC tapes produced from any of the relevant utilities
- other manufacturers tape conventions

Similarly floppy disk file conventions can be adopted from those already in existence.

4.4.2 ELECTRONIC TRANSFER PROTOCOLS

SFDU data format definitions typically appear as text files (ASCII), and SFDU instances typically appear as transparent octet strings. Transfer of individual SFDUs is provided by message level protocols. Transfer of ASCII and "transparent" data files are provided by most available file transfer protocols (FTPs). In the short term, a widely available and inexpensive proprietary FTP may be used (e.g., those used with Decnet, SNA, Telemail, Microcom Networking). In the long term, the ISO FTP and the CCITT Reliable Transfer Protocol are expected to be available and widely implemented.

4.5 SFDU INTERPRETATION

SFDU interpreters may be implemented in different ways. Most will be able to handle only one DDL and only a predefined set of SFDU data format definitions and will interpret SFDU instances in predefined ways. Such interpreters would first verify the SFDU data format definition by interpreting the label.

It is also possible to conceive a "universal" SFDU interpreter capable of handling a wide range of SFDU DDRs.

4.6 ARCHIVES

Archives are facilities which hold data in short- or long-term storage. For SFDU implementations, we are interested in archive interfaces which provide the basic service of sending SFDUs to users in response to queries. The query may be simply a request for a particular SFDU instance or for an ordered set of SFDUs. If SFDUs are not directly available, the SFDU instances need to be generated from the contents of an archive.

The following services may be provided at SFDU user interfaces to archives:

- Accept requests from users (e.g., via public data network or physical mail), specifying SFDUs required plus delivery specifications.

- Respond to request by sending SFDU instances by electronic means or by physical transfer of storage media.

- Allow interactive query of data holdings, i.e., access to catalogues.

- Provide SFDU data format definitions on request. (This latter service would have to be added to existing archives, or it could be provided by a new data dictionary facility.)

The format of data in the archive is not of concern to the user. CCSDS does not propose that the query language for SFDU requests should be standardized at this time.

4.7 DATA DICTIONARIES

An SFDU data format definition is an expression in a DDL. A primitive data dictionary service for SFDU users would be to respond to a request for a data format definition by providing the DDL expression. The DDL expression would itself be formatted as an SFDU instance containing text.

Other data dictionary and related concepts include:

- combining the definition with the data in one SFDU instance (the definition could be text in a designated supplementary label)

- providing references to data dictionaries and component data definitions as part of hierarchical data definitions

ANNEX A

GLOSSARY OF SFDU TERMINOLOGY

ANNEX A - GLOSSARY OF SFDU TERMINOLOGY

ABSTRACT SYNTAX: identifies all the components of a data unit and describes their conceptual structure.

ARCHIVE: a place in which data are preserved. Typically, an archive will provide services for storing and cataloging collected data, and for retrieving selected data upon request. Data delivered to and received from space-related data archives should be in the form of SFDUs, although the data may be stored internally in some other form.

CLOSED DATA SYSTEM: a data system that employs standards applicable at a service access port that are only expected to be known to agents within the enterprise.

CONCEPTUAL STRUCTURE: the organization of a data object/unit used for analytical or deductive reasoning about the information it carries. A conceptual data structure may have several formats, e.g., on different media.

CONCRETE SYNTAX: physical representation.

CONTROL AUTHORITY (CA): a collection of CCSDS member Agency organizations (Control Authority Offices) under the auspices of the CCSDS Secretariat, responsible for registering, archiving, and distributing the data descriptive record (DDR) upon request. Each Agency organization has agreed to meet the minimum recommendations of CCSDS on CA operations.

DATA DESCRIPTIVE LANGUAGE (DDL): a formal notation for specifying the conceptual structure of data objects.

DATA DESCRIPTIVE RECORD (DDR): a set of DDL statements that convey the information necessary to parse the VALUE field of a specific TLV Object.

DATA DICTIONARY (DD): a system that contains the definitions and supplementary information which describe DDRs or data elements/objects.

DATA ELEMENT: the smallest named item or items of data for a given application.

DATA OBJECT: a collection of data elements that are aggregated for or by a specific application.

DATA SYSTEM: a system of an enterprise with the goal to provide services which satisfy the information needs of the enterprise. Major operating facilities of the data system are: physical storage, data management, data retrieval, and data manipulation facilities.

DATA UNIT: an aggregation of data objects which forms a single data interchange entity.

FIELD: an abstract component of a data unit that is assigned a length and a representation. The instance of a field is the field value. Fields may be divided into sub-fields.

FORMAT: the assignment of each of the data elements of a data object to a field or sub-field and to a specific location or address on a given physical medium or in a device.

GLOBAL: pertaining to the CCSDS sphere of influence.

INSTANCE: a data object, or a set of data objects, that exhibits the distinguishing characteristics of its class. An instance of an SFDU is one of a set of values for the data unit which is presented in a specified format. The term instance is used where an individual data unit must be distinguished from its format.

INTERPRET: to explain or present in understandable terms. SFDU interpretation is the process of recognizing the format of a data unit, identifying its component parts, and extracting and presenting the information it carries.

LOGICAL REPRESENTATION: the assignment of data type and data structure attributes to the entities specified to a machine (computer) for a given application. Examples of data type attributes are real, integer, double precision, complex, logical (Boolean), and character. Examples of structural attributes are scalar, array, fields, and logical records.

OCTET: a data object consisting of eight bits.

OPEN DATA SYSTEM: a data system which offers its service to customers outside the enterprise. These outside customers need to know how to operate a service access port. Thus they must be able to: (a) communicate with the system, (b) request data services, (c) accept data products, and (d) elaborate the structure of these products according to standard protocols and structuring rules defined in the public domain.

PHYSICAL REPRESENTATION: the assignment of coded addresses to the data structures placed on storage media or sent through communication networks and the logical representation of the elements that comprise these structures. In communications the address specifies the datum's temporal position. The codification may vary from device to device and medium to medium.

SERVICE: work performed for the benefit of others. Numerous services are involved in the operation of open data systems, including: data collection, conversion, and storage; communication services; archive services such as catalog queries and data distribution; and control authority services such as registration and distribution of (DDRs).

SFDU: Standard Formatted Data Unit. SFDUs are data units that conform to CCSDS recommendations for structure, construction rules, and field specification definition.

TLV OBJECT: the fundamental structural data object that is used to build SFDUs. This object consists of a TYPE field, followed by a LENGTH field, and this is followed by a VALUE field. The flexibility of the VALUE field permits it, under certain conditions, to contain complete TLV Objects and/or complete SFDUs as part of its structure.

ANNEX B

LIST OF ACRONYMS

ANNEX B - LIST OF ACRONYMS

ADI	Authority and Data Descriptive Record Identifier
ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
CA	Control Authority
CCITT	International Telegraph and Telephone Consultative Committee
CCSDS	Consultative Committee for Space Data Systems
DBMS	Data Base Management System
DDL	Data Descriptive Language
DDR	Data Descriptive Record
DEC	Digital Equipment Corporation
ESA	European Space Agency
FTP	File Transfer Protocol
GSFC	Goddard Space Flight Center
ID	Identifier
IBM	International Business Machines
ISO	International Organization for Standardization
JPL	Jet Propulsion Laboratory
NSSDC	National Space Science Data Center
OSI	Open Systems Interconnection
SDIS	Standard Data Interchange Structures
SFDU	Standard Formatted Data Unit
SNA	System Network Architecture
TLV	Type-Length-Value
WDC-A-R&S	World Data Center A for Rockets and Satellites

ANNEX C

EVOLUTIONARY IMPLEMENTATION OF SFDU CONCEPTS

ANNEX C — EVOLUTIONARY IMPLEMENTATION OF SFDU CONCEPTS

1. INTRODUCTION

A limited set of operational scenarios is presented in this Annex. The activities described are neither complete nor the only ones possible to support the operational requirements. It is assumed that a CA mechanism exists, and that this mechanism is responsible for registering, archiving, and distributing format definitions. A more extensive discussion of the CA mechanism will be found in the Control Authority Procedures Book when it is issued.

Section 2, "Initial Operations" is provided to describe a minimum SFDU implementation and uses the exchange of data between two investigators as an example. The environment described could exist a few months after a CA is established.

Section 3, "Intermediate Operations" is provided to describe an SFDU implementation that includes the use of a DDL and gives two examples: (l) exchange of data between two investigators, and (2) receipt and management of data in an archive. The environment described is an update from "Initial Operations" and assumes general purpose software for DDL usage has been created.

Section 4, "Advanced Operations" is provided to describe an SFDU implementation that includes sophisticated operations on SFDUs, and uses Space Station Information System Operations as an example.

2. INITIAL OPERATIONS

Environment

CAs have been established at GSFC and JPL, with minimum guidelines from CCSDS through its agent WDC-A-R&S, located at NSSDC/GSFC. A concrete label structure, and one or more concrete extended structures, have been published and recommended for use. Minimum requirements for the definitions of a data structure, and its content, have been defined. These cover the identification of elements within the data structure, including a one-line description of each element, the units of each element (as appropriate), the representation used for each element, definitions of the logical relations among the elements, the physical locations of each element in the structure and a brief description of the data structure. This information has been described on paper using text and FORTRAN format field descriptors, and submitted to a CA for registration. This combination of text and FORTRAN format description has been used as a primitive DDL, in which the semantic information contained by the text is not machine usable.

A large collection of data, structured according to the format definition, has been stored on magnetic tapes in physical record blocks and within file marks that are not part of the format definition. No SFDU encapsulation structure has been written onto the tapes. An associated piece of software has been written by a local investigator for the machine associated with the data storage. This software has the ability to read a selected volume of data, construct the appropriate SFDU encapsulation bits (not including any format information except the ID), and write the total block as a physical SFDU record without any operating system-specific control bits.

A second piece of software has been written to read physical records from tape, recognize the SFDU structure and format ID, and provide the user with various options:

1. Strip the SFDU encapsulation structure and present the data block to a third piece of software, selected on the basis of the format ID, that knows how to process the given format (hard coded) for required applications.

2. Without stripping the SFDU encapsulation structure, present the SFDU to a fourth piece of software, selected on the basis of format ID, that selectively processes the SFDU instance based on specific content (e.g., find all SFDUs corresponding to a specific time period and copy them to another tape or output device).

The CA has been maintaining a paper file of format definitions and the source of any software, known to it, that works on a given format. A synopsis of each format (including data structures supported) has been maintained on a database, available through dial-up or network access. A compendium of all format definitions has been published and is planned to be updated regularly.

Operations: Investigator-to-Investigator

Investigator A, with machine A, requests a set of data from investigator B, who uses machine B. Investigator B informs A that he will provide the data on unlabeled tapes written in SFDU blocks. He will also send a copy of the format definition, as registered with the CA, and copies of his interpreter software written in FORTRAN 77. Investigator A receives the tapes, documentation, and software and makes an analysis of the best approach to generating operational software on his machine: (l) modify the I/O and word-dependent code of B's software to run on machine A, or (2) use the format definition and generate new code from scratch. Eventually he obtains operational software with the ability to strip the SFDU shell and present the data to his application programs.

Investigator A, although very busy, is supportive of the SFDU approach, and therefore he notifies the responsible CA of the existence of software compatible with machine A to unpack data provided as SFDU structures with the specified format. He is willing to make this software available to others, and to the CA.

Assessment: Investigator-to-Investigator

This scenario does not address all the significant, potential problems (e.g., bit/byte position conventions), but it is useful in highlighting many of them. The major, and perhaps only, advantage of using SFDUs in this manner is the existence of a well-defined, and preserved, format definition. However, this advantage should not be underestimated.

The problem of software development, validation, maintenance, and distribution is readily apparent. This involves software operating on various machines under different operating systems. No CA has access to all the necessary machine types. WDC-A-R&S/NSSDC could archive and distribute the software, but could not guarantee that it was operational.

There is a great potential for format registration explosion, at least until there is a known body of software and a publication of format definitions is supported.

3. INTERMEDIATE OPERATIONS

Environment

CAs have registered a format ID that corresponds to a standard SFDU encapsulation structure using a DDL to completely describe the data content. In response to a request for information on this format, the CA can provide documentation describing the SFDU encapsulation structure and the DDL, and general purpose software to pack and unpack data into or out of this structure. The actual data content, including units, is not known until an SFDU instance is available for examination. The software is available for a number of machines, having been provided by cooperative (actual) users.

Some large collections of data have also been stored on optical disks, in a structure which is not an SFDU instance, but in some cases contains SFDU structures (instances). (Note: At the time of writing, optical disk storage structures have not been standardized with respect to operating system access. This must happen if optical disks are to be exchanged in the future with at least as much compatibility as magnetic tapes are today. It is assumed that there exists sufficient hardware compatibility to make optical disks a reasonable medium for data exchange in this environment.)

The CA has put all machine-readable format definitions into a database that is searchable in various ways (browse capability). These format definitions can be downloaded to requester machines under requester initiative.

Operations: Investigator-to-Investigator

Assuming that investigator B has obtained the software to write his SFDUs in a structure using a DDR and an approved DDL, he informs investigator A of the appropriate format ID and suggests he contact the CA to obtain the software for the A machine. Investigator A does this and readily brings up the interpreter software on his machine. Investigator B sends the magnetic tapes, or a compatible optical disk, or a networked file of SFDU data blocks, to A for processing. Investigator A writes an interface program to select, from the general purpose SFDU/DDL unpacking package (interpreter), the data needed and pass it to his application program. This is much less software than he had to write previously, and it can probably be used for other data as well.

Assessment: Investigator-to-Investigator

A major assumption concerning the availability and utility of general purpose interpreter software for various machines had to be made. This implies that the development effort to obtain the interpreter software for various operating systems and DDLs is considerably less than that needed to permit the required data exchanges without the SFDU approach. Limiting the number of approved DDLs will be necessary. Once past this hurdle, there is considerable incentive to use the SFDU approach in this scenario.

Operations: Investigator-to-Archive

Investigator A wishes to send data to an archive (e.g., NSSDC) and contacts the archive to receive instructions. The archive recommends the data be sent as SFDU instance and provides the investigator with minimum content-documentation requirements. It suggests that the first SFDU instance on each physically distinct medium contain a block of text describing the contents of that medium (e.g., brief descriptions of the data set, how many SFDU blocks are on the medium, an upper limit of the block sizes, a list of SFDU Authority/Format IDs, etc.) that it proposes to use to help manage the data. It suggests that individual SFDU instances correspond to blocks of data that the investigator has found convenient to process, and that each block contains an instance identifier such as a date/time stamp.

Investigator A agrees to send his data in SFDU structures because he has the general purpose software using a DDL to do the packaging. His structures will be self-defining through the inclusion of a DDR in the first SFDU instance of each medium. Each physical record block will be formatted as an SFDU instance. He prepares several magnetic tapes in this way, and sends them to the archive with a cover letter indicating maximum physical record sizes, number of tapes, number of end-of-files, recording density, etc. The documentation needed to parse the data and to relate data elements to each other is included in the SFDU instance as a combination of text and format definitions. (Note:

the SFDU instance could also have been transmitted over a network into a storage buffer at the archive with the file name referenced in a cover letter sent by electronic mail.)

The archive receives the set of data and uses its general purpose SFDU/DDL software to display a list of the content of the first SFDU instance. An archive scientist interactively examines the documentation for completeness. When it has been approved, the data becomes a candidate for automatic scanning and parsing to extract the necessary information for entry into the archives online directory and catalogue. The required inventory information is extractable, using the DDL and general purpose unpacking software. The resulting information is entered into and managed by a DBMS Directory/Catalogue system. The data set is physically stored, still in SFDU structure, on magnetic tape or on optical disk. This approach allows the archive to service requests for data elements by simply reproducing the entire set of data (or a subset of the SFDU) in SFDU form or by extracting specific elements from the data set and packaging them in an SFDU structure for distribution.

Assessment: Investigator-to-Archive

There are many important details that have not been addressed in this scenario, and that need to be clarified to ensure SFDU validity. Again, the existence of a DDL and general purpose software is crucial. The potential for greatly facilitated access to the data will also create an environment that supports better documentation, leading to a qualitative improvement in data management.

4. ADVANCED OPERATIONS

Advanced stages of operations with SFDUs involve a number of technologies that are not in common use everywhere today and may not be fully integrated in data systems for some time to come. Nonetheless, it is important to describe a futuristic scenario to explain what may be achieved with a full implementation of the SFDU concept. This story combines several scenarios, including investigator-to-investigator, investigator-to-archive, and investigator-to-remote-instrument data exchange.

Environment:

CA services have reached a mature level of automation. They regularly support ad hoc data dictionary queries about data value ranges, units of measure, and format definitions. Data format definitions can be supplied in any of several standard DDLs and validated software for interpreting SFDUs is available for a wide variety of machines.

Data archives maintain data, internally, in a mixture of SFDU and non-SFDU forms. However, users see only the SFDU form. Non-SFDU data is distributed either by repackaging it as SFDU instance (e.g., to send it over a communications network) or by providing interface software that forms SFDU instance at read time (e.g., to read non-SFDU digital optical disks). In addition to storing and retrieving data, archives regularly create "new" data for users by correlating data from different sources, converting coordinates, calibrating, correcting for aberrations, and resampling data.

Operations:

A scientist, working at home, is trying to repeat an experiment on Space Station, that had produced inconclusive results when it was flown on an earlier Spacelab mission.

The scientist is able to control his (her) instruments and coordinate their operation with other payload and platform subsystems by exchanging messages (in the form of SFDU instances) through a ground-based communications network and the Space Station Information System. This runs very smoothly because all the command and data format definitions are maintained within the data system.

As the experiment proceeds, the scientist begins to develop a theory about what went wrong in the earlier experiments. To confirm his theory and improve the chances of success for his own experiment, he needs to compare some of the results collected during the earlier Spacelab mission. The only trouble is that he has no idea where these data reside or what form they are in.

His first reaction is to send a message (another SFDU instance) to his colleagues at Kalamazoo U. to see if they know anything about the data he needs to look at. This message is routed through the communications network to the University's computer system. It turns out that his friends no longer have copies of any of the data, but they are able to give him some key information (via a return SFDU instance) about instrument identification codes and the time periods when the experiment had operated.

His next step is to contact the ESA SFDU CA at ESA Space Operations Centre in Darmstadt, West Germany, for whatever information they can provide. This contact is made over an interactive data communications link that allows him to browse ESA's catalogue of format definitions. He is quickly able to focus his search for data formats associated with the instruments used in the experiment, and is able to identify the parameters he is looking for. The catalogue also indicates that all data associated with that mission are archived in NASA's NSSDC.

The data, however, are not stored in SFDU form, since they were collected before the SFDU structure and labels had been standardized by the CCSDS. The steps required to transform old data into SFDU instance for exchange are routine operations now at the NSSDC. Within a few hours of the request, the NSSDC completes the retrieval and reformatting operations and sends the information out on the communications network which routes it to our scientist.

By comparing the data collected from the earlier mission with new data he has collected on the Space Station, our friend is able to confirm his hypothesis about what had compromised the earlier experiment. His next steps are to reconfigure his instruments and modify his real-time processing algorithms to avoid the pitfall he has discovered. These tasks are accomplished by a series of command and data exchanges (all via SFDUs) with the Space Station.

Assessment:

We assume that our scientist's new experiment is successful and that his scientific discoveries lead to important scientific insights. Not every possible form of SFDU exchange has been highlighted in this story, but it does convey the idea that SFDUs will fully support all aspects of space data interchange.